Security of Public Key Certificate Based Authentication Protocols

Wu Wen, Takamichi Saito, and Fumio Mizoguchi

Science University of Tokyo, 2641 Yamazaki, Noda, Chiba, Japan 278-8510
{wen,mizo}@imc.sut.ac.jp, saito@yu.is.noda.sut.ac.jp

Abstract. The security of authentication protocols based on public key cryptography depends on the validity of the certificate. It is usually assumed that a well deployed PKI can guarantee the validity of certificates through mechanisms such as CRL or OCSP. In reality, such guarantee is not always assured. This paper describes an attack that exploits this certificate validity weakness and breaks some well-known certificate-based authentication protocols, namely the SSL and the TLS protocol. This attack affects the "named-server" version of both protocols, but is ineffective for the "named-server, named-client" version of both protocols. Along with the attack, we also describe how it was discovered as a result of our ongoing research on analysis of authentication protocols using both logic based and model checking based methods.

1 Introduction

Soon after the public-key cryptography[3] was invented a rather simplistic approach towards how it can be used to provide integrity and security for digital messages was proposed: a yellow page with name and public key pairs is to be distributed just like a telephone book. To make sure names and public keys are securely binded, certificates[7] were introduced. Later on, the concept of a Public Key Infrastructure(PKI) is envisioned so that the validity of all certificates can be verified by the possession of the root verification key. To solve the problem that valid certificates can become invalid in time due to various reasons, a Certificate Revocation List(CRL) is also provided in PKI.

In theory, any principal who believes that his certificate is compromised can revoke that certificate by putting it on the CRL. Principals intending to use certain certificate can check its validity by searching the CRL. More recently, to lessen the problem of having the client to comb through all the CRLs before knowing if a certificate is invalid, Online Certificate Status Protocol(OCSP) is proposed. These measures, however, can not provide absolute guarantee for certificate validity at all times. On the other hand, commercial products such as the popular Netscape and the Explorer browser have incorporated certificate based authentication protocols such as the Secure Socket Layer(SSL) and Transport Layer Security(TLS). The security of public key certificate based authentication protocols in less ideal situations where a certificate may have been compromised needs to be further analyzed.

H. Imai, Y. Zheng (Eds.): PKC 2000, LNCS 1751, pp. 196–209, 2000.

Authentication is usually conducted between two parties A and B. When public key cryptography is used, public key certificate plays a critical role in authentication protocols thus constructed. A certificate $\{A, K_A^+\}_{K_{CA}^-}$ binds a name A of a principal with its public key K_A^+ through signing using the signature key K_{CA}^- of a trusted third party CA, also known as the Certificate Authority. The secrecy of a message M encrypted by this public key, $\{M\}_{K_A^+}$ is guaranteed by the fact that only the principal with the corresponding private key K_A^- can decrypt this message. In addition, the integrity of a message M signed with the signature key K_A^-, $\{M\}_{K_A^-}$, can be verified with the corresponding verification key K_A^+.

Let us first compare the consequences as a result of key compromises in symmetric shared secret based and public key based systems. In a symmetric shared secret based authentication system, such as a password based authentication system, if the password is compromised, and the fact that it is compromised is not known to both parties of the authentication run, the system is considered broken. Similarly, if the private key corresponding to a certain certificate is compromised, and the fact that it is compromised is not known to both parties, the authentication protocol based on the validity of the certificate is considered broken. This paper does not concern such trivial situations.

In this paper, we are concerned with situations in which a certificate of certain principal is *known* to have been compromised, and is revoked immediately. In the password based authentication example, if the password compromise is known, the promised password will be revoked and a new password will be used. The security of the system is then considered restored. Could this be said of the public key certificate based authentication protocols? This paper describes an attack that exploit the "known" compromised certificate situation and render some version of the TLS and SSL insecure. Furthermore, the ultimate aim of many authentication protocols is to establish a master session key between the authenticated parties to conduct further secure communication. For example, the Netscape browser allows its users to authenticate a web server using SSL before the user gives out his credit card number. The master session key thus generated is then used to encrypt sensitive communication such as the credit card number. In this paper we are concerned with the possibility of leaking of this master session key without the private key of the server certificate currently under use.

2 The Attack on TLS Protocol

In this section, we first give a concrete description of the "named-server, anonymous-client" version of the the TLS protocol. This is followed by a problem statement giving the assumptions made about the environment and the problem we are addressing. Finally, detailed attack trace is presented with discussion of its implications.

2.1 The TLS Protocol Description

In the following, we give a brief description of the TLS protocol by listing the messages communicated between the client and the server. The basic mode of the TLS protocol is the "named-server, anonymous-client" version of the protocol, in which only the server, such as an internet shopping mall, is required to have a certificate. In addition to authenticate the server, this protocol also generates a master session key between an anonymous client and a named server. We use the following notations:

C	Client
S	Server
CA	Certificate Authority
T_i	Timestamp generated by principal i
N_i	Random nonce generated by principal i
K_i^+	Public encryption key for principal i
K_i^-	Secret key of principal i
$\{i, K_i^+\}_{K_{CA}^-}$	i's Public key certificate
$\{...\}_{K_i^-}$	Signed by principal i with key K_i^-
$\{...\}_{K_i^+}$	Encrypted with principal i's public key K_i^+

According to [4,2], the six messages of the protocol are as follows:

$$
\begin{aligned}
&C \rightarrow S : (N_C, T_C) && (M_1)\\
&S \rightarrow C : (N_S, T_S) && (M_2)\\
&S \rightarrow C : \{S, K_S^+\}_{K_{CA}^-} && (M_3)\\
&C \rightarrow S : \{N'_C\}_{K_S^+} && (M_4)\\
&S \rightarrow C : \{H(K_{AB}, AB_5, (M_1, M_2, M_3, M_4))\}_{K_{AB}} && (M_5)\\
&C \rightarrow S : \{H(K_{AB}, AB_6, (M_1, M_2, M_3, M_4))\}_{K_{AB}} && (M_6)
\end{aligned}
$$

where

$$K_{AB} = F((N_C, T_C, N_S, T_S), N'_C)$$

and

$$AB_5 = \text{"server finished"} \ , AB_6 = \text{"client finished"}$$

The messages can be summarized as follows:

M_1**:** C sends a timestamp and a nonce to S;
M_2**:** S sends a different timestamp and nonce to C;
M_3**:** S sends its certificate to C;
M_4**:** C returns the "pre-master secret" N'_C encrypted under K_S^+;
M_5**:** S sends a hash of the session key, a tag AB_5 indicating the protocol stage, and all preceding messages sent by S to C, encrypted with session key K_{AB};
M_6**:** C sends a hash of the session key, a tag AB_6 indicating the protocol stage, and all preceding messages sent by C to S, encrypted with session key K_{AB}.

The master session key K_{AB} is also used by the record layer to encrypt all communications from this point on.

2.2 Problem Statement

In the following, we define the problem we are addressing by stating the following assumptions:

Certificate validity assumption:

- A certificate is always verified using the verification key of the CA. It is impossible for a third party to fake such certificate;
- Certificates have unique names. It is impossible for a client to mistake a third party's certificate for that of the server;
- The signature key of the CA is secure;
- CRLs are only published periodically. It is reasonable to assume that a revoked certificate may still be used before the user find out from either the CRL, or the owner of the certificate that it is revoked;
- OCSP is not integrated with TLS, the validity of the certificate is not always checked at the start of the TLS session.

The last two assumptions appear to be artificial. However, CRL's inadequacies have been pointed out by various researchers[5,6]. We would further argue that

- in the CRL scheme, although the integrity of the CRL is assured by CA's verification key, it is not scalable if the client must keep track of all the related CRLs;
- in the OCSP scheme, a secure channel must be established between the enquiring client and the OCSP server to assure its integrity. However, to establish such a secure channel, protocols like TLS are needed. This leads to a chicken-and-egg scenario;
- if TLS includes certificate status information in message M_3, before a secure connection is established between the server and the client, such status information may not be trusted.

It is apparent from the above arguments that there is no easy way for a client to have correct and timely certificate status information at all times. We further assume the circumstances of a compromised certificate as follows:

Compromised certificate assumption:

- a third party has obtained a copy of the private key that corresponds to the server's certificate;
- the owner of the certificate is aware of the compromise;
- the owner revokes the certificate and list it on the CRL,
- and the owner obtains a new certificate from the CA.

The above assumptions describe a plausible scenario in today's internet-based use of certificates such as those involving SSL with Netscape and Explorer

browsers. For example, if a webmaster for an on-line banking site is fired, the bank will assume that the corresponding certificate is compromised. If the bank is sensible, it is likely to revoke the certificate and obtain a new certificate for future use.

If we ignore the trivial case where the ex-webmaster simply sets up a fake website to impersonate the on-line banking site, we may intuitively argue that the secrecy requirement of TLS seems to be satisfied. Let us assume that the intruder I replaces message M_3 in the TLS protocol, described in section 2.1, with the compromised certificate. Upon receiving the compromised certificate, C either find out from CRL that it is compromised, or send M_3 using the public key contained in the compromised certificate. On receiving M_4, S will find out that it is unable to decrypt it since he now has a new secret key corresponding to the new certificate. The protocol run will be aborted and C will then obtain the new certificate. It may be argued that since N'_C is just a random number, S will simply decrypt M_4 without noticing it was encrypted with a different key. In that case, the calculated master secret K_{AB} will be different for C and S and the protocol will be aborted after message M_5. In this case, the TLS server is in fact also acting as an OCSP server in showing that the certificate used by the client is invalid. In the following we show such arguments do not stand.

2.3 How to Stealing the Master Secret

In the following, we describe the steps showing how an intruder with a compromised certificate can learn the pre-master secret, and therefore the master session key, even if the server has revoked and updated its certificate. Message in the form of "... $\rightarrow (X)I$" indicates that a message intended for X is intercepted by I. Message in the form of "$(X)I \rightarrow$ ..." indicates a message faked by I as from X. Note that $\{S, K'^{+}_S\}_{K^-_{CA}}$ is the compromised certificate of S while $\{S, K^+_S\}_{K^-_{CA}}$ is the fresh and valid certificate of S. M'_i and M''_i are messages that are intercepted and faked by the intruder respectively. They are identical in format with M_i otherwise.

$$
\begin{array}{lll}
C \rightarrow S: & (N_C, T_C) & (M_1)\\
S \rightarrow C: & (N_S, T_S) & (M_2)\\
S \rightarrow (C)I: & \{S, K^+_S\}_{K^-_{CA}} & (M'_3)\\
I(S) \rightarrow C: & \{S, K'^{+}_S\}_{K^-_{CA}} & (M''_3)\\
C \rightarrow (S)I: & \{N'_C\}_{K'^{+}_S} & (M'_4)\\
(C)I \rightarrow S: & \{N'_C\}_{K^+_S} & (M''_4)\\
S \rightarrow (C)I: & \{H(K_{AB}, AB_5, (M_1, M_2, M'_3, M''_4))\}_{K_{AB}} & (M'_5)\\
(S)I \rightarrow C: & \{H(K_{AB}, AB_5, (M_1, M_2, M''_3, M'_4))\}_{K_{AB}} & (M''_5)\\
C \rightarrow (S)I: & \{H(K_{AB}, AB_6, (M_1, M_2, M''_3, M'_4))\}_{K_{AB}} & (M'_6)\\
(C)I \rightarrow S: & \{H(K_{AB}, AB_6, (M_1, M_2, M'_3, M''_4))\}_{K_{AB}} & (M''_6)
\end{array}
$$

In the following we briefly explain the meaning of some of the above messages.

– M_1 and M_2 are sent in plain text;

- In M_3' the valid certificate for S is intercepted;
- The intruder substitutes it with the compromised certificate;
- In M_4', the intruder intercept the message and learns N_C', the pre-master secret;
- The intruder then fake M_4' using the public key contained in the valid certificate;
- Since the intruder knows N_C, T_C, N_S, T_S and N_C', it is able to calculate the master session key K_{AB};
- The interception and faking of the last four messages are now possible and required since both C and S must maintain a consistent record of the past messages despite the different versions of M_3 and M_4 kept by C and S respectively.

Note that messages M_5 and M_6 were designed to prevent attacks that are based on interception and faking of messages $M_1, \cdots, M_4$. For example, if M_5' is allowed to reach C, C will find out that the hash of all messages doesn't match because C expects (M_1, M_2, M_3'', M_4') but instead receiving (M_1, M_2, M_3', M_4''). We can see from above that it is ineffective in preventing the interception and faking described above because the master session key K_{AB} is known to the intruder. After the verification steps M_5, M_6, a TLS session is regarded securely established between C and S. Unfortunately the master secret is now known by a third party. Credit card number and pin numbers that are sent over this TLS session is now known to the third party, an ex-webmaster for example.

2.4 Discussion

TLS is derived from SSL. In the "named-server" version of the SSL protocol, identical attack as describe above also leads to the leaking of the the master session key K_{AB}. However, in the "named-sever, named client" version of both SSL and TLS protocol, the above attack doesn't succeed. This is explained next.

In the "named-sever, named-client" version of the TLS protocol, both clients and server are authenticated. In message M_4 of this version of the protocol, C's certificate and a signed hash of the list of previous messages so far transmitted, in addition to the pre-master secret encrypted with S's public key, are sent by C to S. This is shown as follows:

$$C \rightarrow S : \{C, K_C^+\}_{K_{CA}^-}, \{N_C'\}_{K_S'^+}, \{..., \text{Hash}(M_1, M_2, M_3''), ...\}_{K_C^-}(M_4)$$

The parts shown as "..." are tags and other parameters such as the master secret computed by C at this time and are not important for the following discussion. Again, the intruder I can learn the pre-master secret N_C' using $K_S'^-$, and compute the master secret K_{AB}. It is however, unable to fake the signed portion of M_4 because it does not have the signature key K_C^- of the client C. In order to maintain the consistency of the list of transmitted messages at both C and S side, the signed part, as well as the encrypted pre-master secret in M_4, must also be altered. Due to the inability to alter the signed portion of M_4 by the intruder, it is possible now to detect the interception and faking of M_3 and M_4 in step M_5 or M_6. The protocol run will be aborted and the attack will fail.

3 How Were the Above Attacks Discovered?

We have been interested in using both model checking method[14] and logic checking method[13] for analyzing authentication protocols. First we extended the BAN logic[1] to allow it to deal with the analysis of secrecy theorems and succeeded in discovering a weakness in the fix[8] proposed by Gavin Lowe for the Needham-Schroeder public key authentication protocol. Model checking is then used to discover the exact attack for the described weakness. It turned out that the weakness discovered in the Needham-Schroeder protocol affects the "named-server" versions of both SSL and TLS.

The next two sections describe in summary the results of our analysis of the Needham-Schroeder authentication protocol using both methods. Readers interested in the details should consult [13] and [15].

4 Extending BAN Logic for Secrecy Analysis

BAN logic was proposed to reason and analyze properties of authentication protocols such as Kerberos, Needham-Schroeder, and SSL etc. A number of weaknesses in various protocols were discovered using BAN logic. However, due to its modeling of the principals as benign agents, as pointed out by Dan Nessett in [10], it is considered ineffective in analysis of secrecy related properties of authentication protocols. As described in [10], the environment modeled in BAN does not include principals that are not supposed to gain access to certain secrets. Since such principals are not modeled, it is therefore impossible to analyze property related to secrecy. Nessett further illustrated this weakness by describing an obviously flawed protocol and proving proper authentication using the BAN logic.

The criticism is somewhat unfair in that, the designer of BAN logic has never claimed it can be used to deal with secrecy. Only proper authentication, as defined by a set of beliefs, were defined and used in the analysis.

Our work is partially motivated by this criticism, as well as other recent research in dealing with the analysis of secrecy properties of authentication protocols. For example, Paulson has incorporated secrecy theorems into his logic for security protocols in [11]. Roscoe described a model for a "spy" in [12]. Lowe has successfully discovered an attack[8] on Needham-Schroeder public key authentication protocol using a model of the protocol including an intruder. Other results on verification of SSL[9] and TLS[4,11] also encouraged our investigation.

4.1 Parameterization of BAN Logic

In the following, the Needham-Schroeder public key authentication protocol is described briefly before the parameterization is explained using this protocol as an example. We will refer to this protocol as the original protocol from now on. In the simplified version of this protocol, only two principals, an initiator A and an responder B are considered. Three messages are exchanged:

$$A \rightarrow B : \{N_A, A\}_{K_B^+} \quad (M_1)$$
$$B \rightarrow A : \{N_A, N_B\}_{K_A^+} \quad (M_2)$$
$$A \rightarrow B : \{N_B\}_{K_B^+} \quad (M_3)$$

We introduce a third principal, an intruder I and model the above protocol using parameterization. First we fix the initiator A, then we parameterize the responder by a variable X_B, where $X_B \in \{B, I\}$, This allow us to model the protocol run that includes possible interception and faking of messages by the intruder. For example, the following message:

$$A \rightarrow X_B : \{N_A, A\}_{K_{Y_B}^+}, \text{where } Y_B \in \{B, I\},$$

can be interpreted as

- $X_B = Y_B = B$: a message sent from A to B encrypted with B's public key, which corresponds to the normal message described in the original protocol definition;
- $X_B = Y_B = I$: a message sent from A to I using I's public key;
- $X_B = B, Y_B = I$: a message sent from A, intended for B, using I's public key.

Note that we use Y_B and X_B to indicate that the intended receiver and its public key bearer may be different.

4.2 Secrecy Property in Parameterized BAN

With the introduction of an intruder I, we are in the position to define secrecy property for the above protocol.

Definition 1. *If both shared secret N_A and N_B, intended for an authentication session between A and B, can be obtained by intruder I, the secrecy property of the protocol is considered violated. In terms of the parameterized BAN logic, if the formulas corresponding to "I saw N_A" and "I saw N_B" can be derived from the protocol, we regard the secrecy of the protocol violated.*

For the original Needham-Schroeder protocol described above, both formula corresponding to "I saw N_A" and "I saw N_B" can be derived[13]. This indicates that the secrecy property for the protocol is violated. Unfortunately, the analysis does not directly point to an attack. In [8], Lowe discovered the following attack using model checking method. We will refer to this attack as the "impersonation attack".

$$\alpha : A \rightarrow I \quad \{N_A, A\}_{K_I^+}$$
$$\beta : (A)I \rightarrow B \; \{N_A, A\}_{K_B^+}$$
$$\beta : B \rightarrow (A)I \; \{N_A, N_B\}_{K_A^+}$$
$$\alpha : I \rightarrow A \quad \{N_A, N_B\}_{K_A^+}$$
$$\alpha : A \rightarrow I \quad \{N_B\}_{K_I^+}$$
$$\beta : (A)I \rightarrow B \; \{N_B\}_{K_B^+}$$

The above trace describes two interleaving runs α and β. Run α is between A and I, and run β is between $(A)I$ and B. $(A)I$ indicates intruder I impersonates A. Lowe proposed a fix in [8] to deny this attack. In the fixed protocol, message 2 is

$$B \rightarrow A : \{N_A, N_B, B\}_{K_A^+}$$

We will refer to this addition as "Lowe's fix". This prevents the "impersonation attack" since message 2 can not be simply copied from run β to run α. A will expect I rather than B as its corresponded. This fix, however, is shown ineffective to the "ex-webmaster attack" described in the section 5.4.

4.3 Security of the Fixed Needham-Schroeder Protocol

We idealize message 2: $\{N_A, N_B, B\}_{K_A^+}$ in the fixed Needham-Schroeder protocol by adding the substitution term: $X_B = B$ to our model of the Needham-Schroeder protocol using parameterized BAN logic. In this case, the inference results does not contain the formula "I see N_B". Formulas corresponding to proper authentication described in the original BAN logic can also be obtained.

This result appears to indicate that the fixed Needham-Schroeder is secure. However, on closer examination, the secrecy property is held if the assumption $X_B = B$ is guaranteed. The security of the fixed protocol is therefore dependent on this assumption. This turned out to be the weakness of the fixed Needham-Schroeder protocol.

Quick discussion: Analysis based on logic, such as the BAN logic, does not directly point to possible attacks. Rather, it highlights the weakness in the protocol by pointing to certain weak assumptions that may not stand in reality. Further analysis of such assumptions may lead to the discovery of actual attack. Our study also follows this pattern. In the next section, we show how an attack on this fixed protocol can be discovered.

5 Discover Concrete Attacks Using Model Checking

Unlike logic based methods such as the BAN logic, in which messages are abstracted and idealized to logic formulas representing certain beliefs held by each principals, model checking method requires a concrete state based model to be constructed. Usually, each principal is represented by a process. States of the process such as "running" or "committed" are reached depending on the messages that are sent and received. If a specification described as an assertion or a temporal logic formula is shown to be untrue with respect to the model, the model checker also provides a trace of the actual attack.

In the following, we describe the state based model of the protocol that includes an initiator, a responder and an intruder. The secrecy requirements used to discover the "ex-webmaster attack" is also described.

5.1 State-based Model of Needham-Schroeder Protocol

The state-based model consists of three concurrent processes representing the initiator, the responder and the intruder. The initiator and the responder send and receive messages strictly according to the protocol specification. The intruder, however, are allowed the following behavior:

- overhear and store a copy of the message;
- steal/intercept a message from its intended receiver;
- decrypt message that are encrypted with its public key;
- replay any stored message at any time;
- make up new messages using learned secret such as stolen nonces.

The above assumptions are based on those describe in [8] and are justified in the current Internet environment assuming that cryptography used by the protocol is strong enough. An illustration of the model is shown in Figure 1.

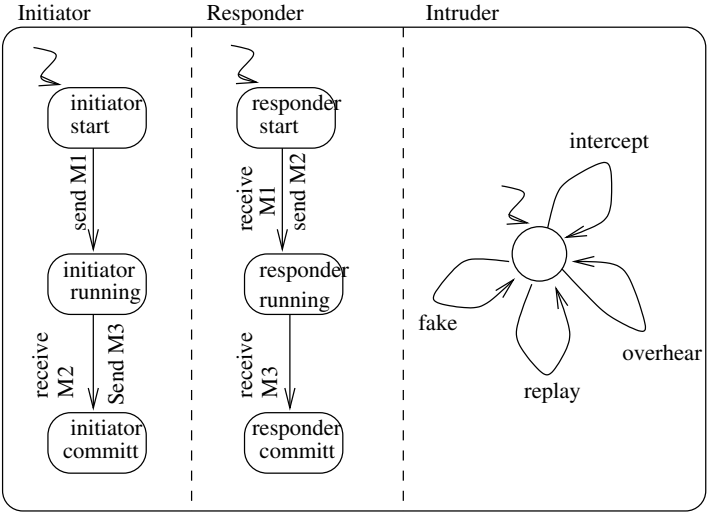

Fig. 1. The state-based model for Needham-Schroeder Protocol

Based on the results describe in section 4.3, it is clear that the security of Lowe's fix depends on whether we can guarantee $X_B = B$ when receiving message 2. This is assumed in Lowe's analysis. To further explore the consequences of possible certificate compromises we deliberately weaken this assumption and allow the possibility that A may not have the means to positively identify X_B as B. In other words, it is possible for A to send a message intended for B using public key that may or may not be that of B's. This is to say that message of the following format

$$A \rightarrow B : \{N_A, A\}_{K_I^+}$$

is not excluded from our model.

This is the only difference between our protocol model and that of Lowe's. We will give an justification for weakening this assumption in section 5.4.

5.2 Nonce Secrecy Requirements

In [8], the requirements for proper authentication are described as safety properties:

- If the responder is committed to the initiator, the initiator must be running with the responder;
- If the initiator is committed to the responder, the responder must be running with the initiator.

It is quite clear that such requirements are concerned with the possibility of the intruder impersonating either the initiator or the responder.

Our results of analysis given in logic checking phase, however, points to the necessity of protecting the secrecy of the nonces. Consequently, our requirements are given as a safety property of the secrecy of the nonces N_A and N_B.

- Intruder I must not be able to learn both N_A and N_B in protocols runs between A and B.

5.3 Results of Model Checking

Two versions of the Needham-Schroeder protocol were model checked using the secrecy requirement described in the previous subsection: the original Needham-Schroeder and the version with Lowe's fix.

Both the "impersonation attack" and the "ex-webmaster attack" are found in the results of model checking the original version of the protocol. Only the "ex-webmaster attack" was discovered in the version with Lowe's fix. The traces of the attacks are shown in Figure 2 and Figure 3 respectively. Note that the "impersonation attack" is discovered using the secrecy requirement rather than the proper authentication requirement used by Lowe. In fact, Lowe's attack can also be described as the result of I learning N_A, N_B, which is supposed to be a shared secret between A and B.

The model checking result for the version with Lowe's fix only produced the attack corresponding to the "ex-webmaster attack". In other words, Lowe's fix is effective towards the "impersonation attack" but not effective towards the "ex-webmaster attack". The author can verify this by modifying message 3 in Figure 3 and see that it has no effects in preventing the "ex-webmaster attack".

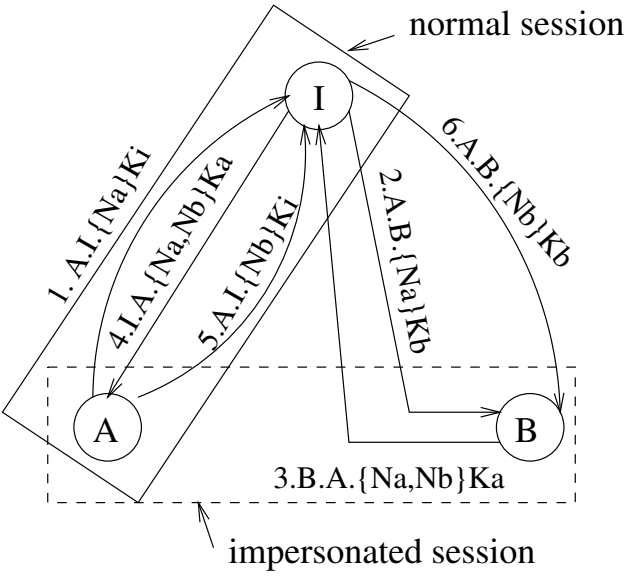

Fig. 2. The "impersonation attack" due to Lowe

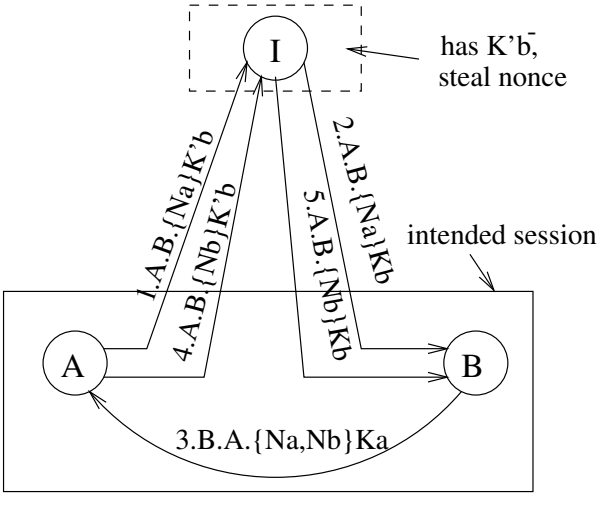

Fig. 3. The "ex-webmaster attack" due to Wen-Saito

5.4 Interpretation of the "Ex-Webmaster Attack"

On first glance, it might be difficult to imagine that A might, by its own will, send an encrypted message addressed to B, using the public key that belongs to the intruder. Closer analysis shows that such situation is quite possible. As described in section 2.2 as the attack on the "named-server" version of both SSL and TLS protocol, the client C sends the pre-master secret N'_C using the compromised public key of the server S. We can easily suggest that the compromised public key of the server, is in fact the public key of the intruder described in here.

5.5 Another Fix to Needham-Schroeder

We propose a fix that can eliminate both "impersonation attack" and the "ex-webmaster attack" on the public key version of the Needham-Schroeder authentication protocol. Instead of adding only B to message 2 in the original protocol, we propose to add K_B^+. The fixed protocol is as follows:

$$\begin{aligned} A \rightarrow B &: \{N_A, A\}_{K_B^+} && (M_1) \\ B \rightarrow A &: \{N_A, N_B, K_B^+\}_{K_A^+} && (M_2) \\ A \rightarrow B &: \{N_B\}_{K_B^+} && (M_3) \end{aligned}$$

Since the intruder I is unable to decrypt message 2 and replace K_B^+ with $K_B'^+$, A will be able to detect the inconsistency. Even if it chooses not to check for inconsistence of K_B^+ and $K_B'^+$, message 3 will be encrypted with K_B^+, the new and valid public key, and the intruder will be unable to learn nonce N_B from message M_3.

6 Discussion

When analyzing certificate based protocols, it is normally assumed that the validity of the certificates is guaranteed by the deployed PKI. In reality, methods for guaranteeing such validity, such as CRL and OCSP, are either ineffective, or difficult to be incorporated into the various authentication protocols used currently. The case of total lose of security as the result of a compromised certificate, where the owner is not aware, does not seem to have a solution. If, however, one of the principals engaged in the authentication run has a compromised certificate, knows about it, revokes it and replaces it with a new certificate, it appears that the protocol should be safe since the owner will detect such inconsistency and stop the run. The results described in this paper shows that an attack taking advantage of the "known" invalid certificate exists for the "named-sever" version of the most popular authentication protocols used today.

The ineffectiveness of the attack towards the "named-server, named-client" version of both SSL and TLS suggest that, signing the hash of all previous messages by the client has its merit.

Unfortunately, the fix we proposed for the Needham-Schroeder public key authentication protocol is unsuitable for fixing SSL and TLS. Since the client

certificate is not used in the "named-sever" only version of SSL and TLS, it is not possible to securely pass the new and valid certificate in the current protocol run. If client C's certificate is used, as in the "named-server, named-client" version of the protocol, signing the hash of all previous messages by the client should also prevent this attack.

References

1. M. Burrows, M. Abadi, and R. Needham. A logic of authentication. Technical Report 39, DEC Systems Research Center, February 1989. 202
2. T. Dierks and C. Allen. The tls protocl: Version 1.0. Technical Report drat-ietf-tls-rptocol-05.txt.Z, IETF task force, May 1998. 198
3. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976. 196
4. Walter D. Eaves. Transport level security: a proof using the gny logic. Technical report, Brunel University, UK, February 1989. 198, 202
5. C. Ellison. Can we eliminate crl? Technical report, extended abstract, November 1997. 199
6. C. Ellison and et al. Spki certificate theory, internet draft. Technical report, IETF SPKI Working Group, November 1997. 199
7. L. Kohnfelder. Towards a practical public-key cryptosystem. Technical report, MIT, 1978. 196
8. G. Lowe. Breaking and fixing the needham-schroeder public key protocol using csp and fdr. In *TACAS96*, 1996. 202, 203, 204, 205, 206
9. J. Mitchell, M. Mitchell, and U. Stern. Automated analysis of cryptographic protocols using murphi. In *IEEE Symposium on security and privacy*, pages 141–151, 1997. 202
10. D. Nessett. A critique of the burrows, abadi and needham logic. *ACM Operating Systems Review*, 24(2):35–38, April 1990. 202
11. L. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6:85–128, 1998. 202
12. A. W. Roscoe. The perfect 'spy' for model-checking crypto protocols. In *DIMACS Workshop on Design and Formal Verification of Security Protocols*, Piscataway, NJ, September 1997. 202
13. T. Saito, W. Wen, and F. Mizoguchi. Analysis of authentication protocol by parameterized ban logic. Technical report, ISEC, July 1999. 202, 203
14. W. Wen and F. Mizoguchi. Model checking security protocols. In *Symposium on Cryptography and Information Security*, pages 647–652, Kobe, Japan, January 28–29, 1999. 202
15. W. Wen, T. Saito, and F. Mizoguchi. New results of model checking needham-schroeder protocol. in proceedings of 1999 Computer Security Symposium, pages 69–74, Kanazawa, Japan, October 21–22, 1999. 202